

Processo de Desenvolvimento de Software

Escritório de Gerenciamento de Processos de
Tecnologia da Informação e Comunicação – EGPr-TIC

João Pessoa – 2016

Versão 0.4.2

Tribunal Regional do Trabalho da 13ª Região

Desembargador Presidente

Ubiratan Moreira Delgado

Desembargador Vice-Presidente

Eduardo Sérgio de Almeida

SETIC - Secretaria de Tecnologia da Informação e Comunicação

Ronaldo de Araújo Farias (Diretor)

NGTIC - Núcleo de Governança de Tecnologia da Informação e Comunicação

Davi Medeiros Cabral

Marcelo Luis Machado Moura

Samuelson Wagner de Araújo e Silva

Wagner de Souza Porto

Coordenadores da SETIC

Daniel Nunes Barbosa

Luis Fabiano Saldanha Bandeira

Rodrigo Cartaxo Marques Duarte

Rogério Nunes Silva

Sumário

1.Objetivo.....	5
2.Escopo.....	5
3.Definições.....	6
4.Políticas e Diretrizes.....	7
5.Fluxo do Processo.....	8
6.Descrição das principais atividades do processo.....	9
6.1.Analisar Demanda.....	9
6.2.Alimentar o Backlog.....	10
6.3.Priorizar e Delegar Tarefa.....	10
6.4.Levantamento de Requisitos e Codificação da Demanda de TIC.....	10
6.4.1 Levantar Requisitos.....	10
6.4.2 Codificar Demanda.....	11
6.5.Revisar Tarefa.....	11
6.6.Preparar Ambiente de Teste e Homologação.....	11
6.7.Realizar Teste.....	12
6.8.Homologar Sistema.....	12
6.9.Atualizar Manual do Usuário.....	13
6.10.Treinar CAU.....	13
6.11.Treinar os Multiplicadores do Conhecimento.....	13
6.12.Treinar os Usuários Finais.....	13
6.13.Implantar em Produção.....	13
7.Papéis e responsabilidades.....	14

8. Indicadores de Desempenho.....	14
9. Sugestões de melhorias futuras.....	17
10. Referências.....	18
11. Histórico de versões.....	19

1. Objetivo

Este processo de Desenvolvimento de Software foi especificado para a criação, melhoria e correção dos sistemas do TRT da 13ª Região.

2. Escopo

O desenvolvimento de software é hoje um assunto presente em organizações que criam e mantêm softwares. Segundo Spinola¹, a vida de um software não termina após a sua implantação. Ele ainda viverá durante muito tempo. Será utilizado por anos, e, com certeza, terá muitas atualizações, gerando novas versões do sistema.

Nesse contexto, por criação de software, entende-se a execução de uma atividade inovadora, produzindo uma nova funcionalidade, um novo sistema de informação, ou seja, um sistema que a organização ainda não possui, e que tem como objetivo automatizar um conjunto de atividades realizadas manualmente.

Por outro lado, a manutenção é caracterizada pela modificação do software já entregue ao cliente, ou seja, a manutenção é qualquer alteração ou modificação no software após sua entrada em produção, podendo esta ser de dois tipos: corretiva ou evolutiva².

A manutenção corretiva compreende a correção de erros, defeitos, problemas encontrados durante a utilização do sistema, seja pelos usuários, seja pelos

1 Idem.

2 Idem.

desenvolvedores ao realizarem testes. Já a manutenção evolutiva ou melhoria compreende o aperfeiçoamento das funções de um sistema já existente.

No Tribunal Regional do Trabalho da 13ª Região, esse conceito é aplicável a todos os serviços prestados pela Secretaria de Tecnologia da Informação e Comunicação (SETIC) e pelas coordenadorias de apoio negocial ao Processo Eletrônico (CSP-PJe e CAN-PJe), os quais estão no Catálogo de Serviços de TI.

3. Definições

Neste documento, são adotadas as seguintes definições:

Backlog da CDMS: Sistema interno de demandas mantido e usado apenas pela CDMS.

Banco de dados de erro conhecido (BDEC): Um banco de dados que contém todos os registros de erros conhecidos.

CAU: Coordenadoria de Apoio ao Usuário

CDMS: Coordenadoria de Tecnologia e Suporte Técnico.

CSUAP: Coordenadoria de Desenvolvimento do Sistema Unificado de Administração de Processos.

CTST: Coordenadoria de Tecnologia e Suporte Técnico

Problema: A causa raiz de um ou mais incidentes. A causa geralmente não é conhecida no momento em que o registro de problema é criado, devendo a sua investigação a ser conduzida posteriormente.

Erro/defeito conhecido: Um problema que possui causa raiz e solução de contorno documentadas. Erros conhecidos são criados e gerenciados durante todo o seu ciclo de vida. Erros conhecidos também podem ser identificados pelo

desenvolvimento ou fornecedores.

Melhoria: Demanda por novas funcionalidades para melhorar a aplicabilidade e usabilidade do software, de maneira a ajustar o produto às diversas necessidades da Administração. A melhoria, via de regra, não está atrelada a um erro/defeito.

Demanda Interna: Demanda originada por algum setor, núcleo ou coordenação interno à SETIC.

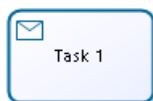
Demanda externa: Demanda originada por algum setor, núcleo ou coordenação externo à SETIC.

Problema: A causa raiz de um ou mais incidentes. A causa geralmente não é conhecida no momento em que o registro de problema é criado, devendo a sua investigação a ser conduzida posteriormente.

OTRS: Iniciais para Open-source Ticket Request System. Sistema de gerenciamento de incidentes livre e de código aberto mantido pela SETIC, para atribuir rótulos para a entrada de requisições e acompanhar comunicações futuras sobre elas. É um meio de gerenciar consultas recebidas, reclamações, pedidos de suporte, relatórios de defeitos e outras comunicações.



: O resultado da atividade será registrado no OTRS.



: A atividade demandará leitura de registro no OTRS.

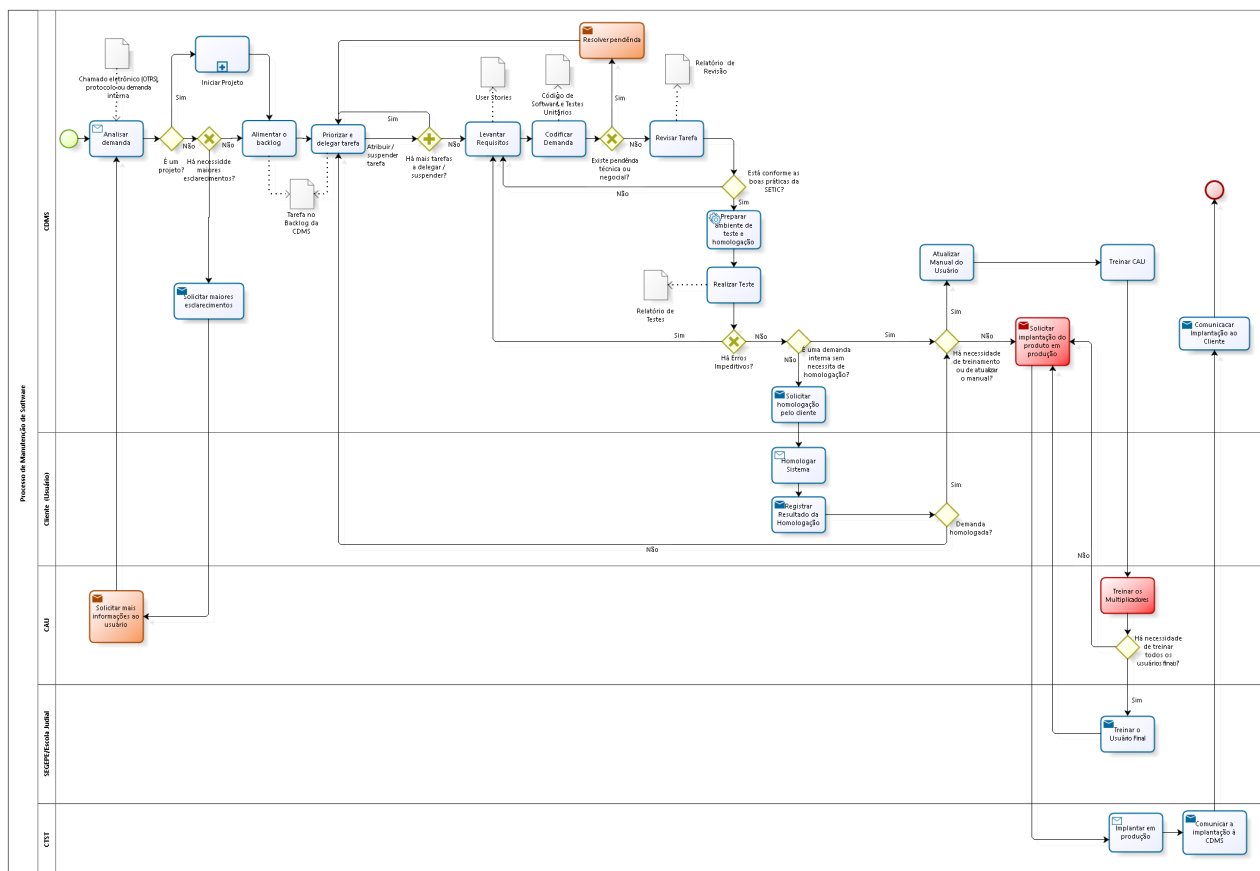
4. Políticas e Diretrizes

O Desenvolvimento de Software em sentido amplo deve estar alinhada às seguintes políticas e diretrizes:

- Todas as criações, melhorias e correções de software decorrerão de demandas previamente registradas e priorizadas;
- A categorização e priorização das demandas de manutenção de Software serão feitas pela CDMS, que tomará como referência o PDTIC e as deliberações da Comissão Permanente de Informática.
- A categorização e priorização das demandas para criação e melhoria de softwares serão feitas Pela Comissão Permanente de Informática e definidas no PDTIC.
- Toda criação, melhoria ou manutenção software que alterar os procedimentos de trabalho, só será concluída com a realização de treinamentos e a alteração do manual do usuário.
- Os estados da aquisição, criação, melhoria ou manutenção software deverão ser comunicados ao usuário, bem como a data prevista para a entrega da solução no ambiente de produção.
- A CMDS deverá solicitar mais informações do usuário quando a demanda não dispuser de informações suficientes para o atendimento.
- O banco de dados de erros conhecidos deve ser atualizado constantemente.

5. Fluxo do Processo

A figura abaixo mostra o fluxo do processo Desenvolvimento de Software.



6. Descrição das principais atividades do processo

6.1. Analisar Demanda

Todo Desenvolvimento de Software será proveniente das demandas negociais registradas previamente no PDTIC e nos sistemas de chamado de eletrônico e do protocolo eletrônico homologados pelo TRT da 13ª Região, bem como de questões técnicas internas à CDMS, como reescrita de código, visando melhorar o desempenho e/ou facilitar manutenções futuras dos softwares mantidos pela SETIC.

Essa atividade tem como objetivo selecionar, com base nas diretrizes do PDTIC e das deliberações da Comissão Permanente de Informática (CPI), quais demandas desencadearão o processo de desenvolvimento de software.

Caso a demanda seja de um novo projeto, isto é, da criação de um novo sistema

ou funcionalidade, serão seguidos os passos do subprocesso “Iniciar Projeto”.

Caso contrário, isto é, seja uma demanda de manutenção de software já existente, e as informações contidas no chamado não sejam suficientes, a CDMS solicitará à CAU maiores esclarecimentos.

Caso não exista mais nenhuma demanda a analisar, será iniciada a atividade de priorização e delegação das tarefas à equipe da CDMS.

6.2. Alimentar o Backlog

As demandas selecionadas a partir da tarefa anterior, Analisar demanda ou Iniciar Projeto, serão registradas no backlog da CDMS ou do Projeto, respectivamente. Somente as demandas registradas neste backlog é que comporão o escopo do desenvolvimento de software gerenciado pela SETIC.

6.3. Priorizar e Delegar Tarefa

Nessa atividade, será priorizada cada demanda, bem como a estimativa de sua entrega e o responsável por sua codificação no software. Tais informações serão atualizadas no *backlog* da CDMS ou do Projeto, a depender do caso, podendo ser revistas a qualquer momento. Dessa revisão, poderá ser gerada a suspensão de uma demanda delegada anteriormente e cuja codificação já tenha sido iniciada, por não mais entendê-la prioritária.

6.4. Levantamento de Requisitos e Codificação da Demanda de TIC

6.4.1 Levantar Requisitos

Nesta atividade, será criado ou revisado o documento *User Stories*, contendo o detalhamento técnico dos requisitos do cliente sobre a demanda priorizada.

6.4.2 Codificar Demanda

Com base nas *User Stories*, os programadores codificarão a demanda, seja ela corretiva ou evolutiva, bem como os respectivos testes unitários, e salvarão ambos artefatos no GIT.

Caso durante a codificação surja alguma dúvida negocial, serão solicitados maiores esclarecimentos ao usuário demandante (atividade Resolver pendência) e, até que estes sejam respondidos, a demanda ficará suspensa.

6.5. Revisar Tarefa

A atividade Revisar Tarefa está atrelada à revisão das alterações com fito a garantir a qualidade do texto das *User Stories*, do código do software modificado e dos *scripts* de banco de dados gerados. As *User Stories* e o código do software serão revisados por um desenvolvedor e o do *script* por um administrador de banco de dados (BDA), ambos da CDMS e que não tenham participado diretamente da implementação da demanda.

Caso a codificação esteja conforme os padrões técnicos da CDMS, será autorizado o *merge request* e, com isso, o código passará a integrar oficialmente a versão de teste e homologação.

Se a codificação estive fora do padrão de qualidade (boas práticas) definido pela SETIC, a demanda voltará para o desenvolvedor que a implementou. Caso este desenvolvedor já esteja atendendo a outra demanda de prioridade igual ou inferior, esta será interrompida para que o código revisado seja corrigido.

6.6. Preparar Ambiente de Teste e Homologação

Após a codificação da mudança, será preparado automaticamente um ambiente, contendo a nova versão do software modificado. Este ambiente será usado tanto pelos desenvolvedores para a realização de testes, quanto pelo cliente da demanda para a homologação da nova versão do software.

A solicitação da atualização do software de teste e de homologação neste ambiente é de responsabilidade da CDMS, enquanto que a manutenção do ambiente em si, ou seja, da infraestrutura de computadores, redes e servidores sobre os quais o software de teste e de homologação funcionará, é de responsabilidade da CTST.

6.7. Realizar Teste

Após a preparação do ambiente contendo o software modificado, desenvolvedores que não tenham participado da codificação desta nova versão realizarão testes com o intuito de identificar possíveis erros e falhas, tanto problemas técnicos quanto negociais, nos termos solicitados das *user histories* escritas a partir da demanda do cliente.

Caso seja encontrado algum erro impeditivo, as mudanças não serão aprovadas e o processo voltará para a atividade “Priorizar e delegar tarefa”.

Caso seja uma demanda interna com necessidade de homologação ou uma demanda externa e não seja encontrado nenhum erro impeditivo, será solicitado ao cliente da demanda que homologue a nova versão do software, sendo o endereço eletrônico do ambiente de homologação fornecido pela CDMS.

6.8. Homologar Sistema

No ambiente informado pela CDMS, o cliente analisará negocialmente as novas funcionalidades implementadas ou as alterações de manutenção feitas no software demandado.

Caso o cliente não esteja de acordo com as alterações feitas no software, ele deverá registrar quais pontos não foram homologados (atividade Registrar Resultado da Homologação) e como deverá ser feita a correção, voltando a demanda para a fase de priorização.

Caso esteja de acordo com as alterações, deverá ser analisada a necessidade de criar ou atualizar o manual do usuário. Havendo esta necessidade, a atualização será feita pela CDMS. Não a havendo, o software segue para sua implantação em produção

pela CTST (atividade Solicitar Implantação em Produção). Caso contrário, a demanda voltará para a atividade Priorizar e Delegar Tarefa.

6.9. Atualizar Manual do Usuário

Após a homologação das alterações de código pelo cliente, caso a interface ou a forma de uso do software tenha sido modificada, serão feitos os devidos ajustes no manual do usuário pela CDMS.

O manual do usuário será armazenado na base de conhecimento da SETIC, usando sistema e formato indicados pelo Núcleo de Governança.

6.10. Treinar CAU

Havendo alterações no manual do software, a CDMS ficará responsável por treinar a CAU sobre tais alterações.

6.11. Treinar os Multiplicadores do Conhecimento

Havendo alterações no manual do software, a CAU ficará responsável pela realização do treinamento dos multiplicadores do conhecimento. Preferencialmente os multiplicadores do conhecimento serão os mesmos que demandaram a nova funcionalidade ou a manutenção do software existente.

6.12. Treinar os Usuários Finais

A Escola Judicial, em conjunto com os multiplicadores do conhecimento, ficarão responsáveis pelo treinamento ao Usuário Final.

6.13. Implantar em Produção

Nesta atividade, a CTST ficará responsável por colocar em produção a nova versão do software, caso a demanda tenha sido homologada pelo Cliente e a implantação tenha sido solicitada pela CDMS.

Após o término desta atividade, a CTST comunicará a atualização do software no

ambiente de produção à CDMS (atividade Comunicar a Implantação à CDMS). Em seguida, a CDMS comunicará o mesmo ao cliente (atividade Comunicar Implantação ao Cliente).

7. Papéis e responsabilidades

Os papéis e responsabilidades dos envolvidos no Processo de Desenvolvimento de Software são definidos conforme a tabela abaixo.

Papel	Responsabilidades
CDMS	<ul style="list-style-type: none">• Codificar as mudanças de software;• Garantir a atualização inicial dos manuais do usuário;• Treinar a CAU em relação às mudanças nos softwares.
CTST	<ul style="list-style-type: none">• Manter os ambientes de teste e homologação;• Disponibilizar, no ambiente de produção, as novas versões dos softwares;• Manter no ar os softwares em produção.
CAU	<ul style="list-style-type: none">• Treinar multiplicadores do conhecimento sobre as mudanças no software.• Manter e evoluir os manuais de software criados pela CDMS.
Escola Judicial	<ul style="list-style-type: none">• Responsável por treinar os usuários finais dos softwares.
Cliente (Usuário)	<ul style="list-style-type: none">• Homologar as novas versões dos softwares;• Demandar mudanças nos softwares mantidos pela SETIC.

8. Indicadores de Desempenho

O Processo de Manutenção de Software será monitorado e constantemente medido através de indicadores de desempenho. Essas medidas serão consolidadas periodicamente pelo Gerente do Processo e farão parte do Relatório Gerencial do processo. Esse relatório tem como objetivo acompanhar a eficácia do processo, identificando tendências, falhas e oportunidades de correções, promovendo sempre a melhoria contínua.

As seguintes métricas foram definidas para o Processo de Desenvolvimento de Software.

1 Taxa de aderência da manutenção à atividade do processo

Descrição	Mede o percentual de aderência das demandas de manutenção que completaram determinada atividade do processo.
Periodicidade	Trimestral
Fórmula	$adrenc_atv_manut(fi) = atv_manut_conc(fi) / atv_manut_total$ <p>Onde:</p> <ul style="list-style-type: none"> fi: Representa uma das seguintes atividades do processo: Codificar Demanda; Revisar Tarefa; Realizar Teste; Homologar Sistema; e Implantar em Produção. adrenc_atv_manut(fi): Aderência das demandas de manutenção em relação à atividade fi. atv_manut_conc(fi): Quantidade de demandas de manutenção concluídas que passaram pela atividade fi. atv_manut_total: Quantidade total de demandas de manutenção concluídas.

2 Taxa de aderência do projeto à atividade do processo

Descrição	Mede o percentual de aderência das demandas de projeto que completaram determinada atividade do processo.
Periodicidade	Trimestral
Fórmula	$adrenc_atv_proj(fi) = atv_proj_conc(fi) / atv_proj_total$ <p>Onde:</p> <ul style="list-style-type: none"> fi: Representa uma das seguintes atividades do processo: Codificar Demanda; Revisar Tarefa; Realizar Teste; Homologar Sistema; e Implantar em Produção. adrenc_atv_proj(fi): Aderência das demandas de projetos em

	<p>relação à atividade fi.</p> <ul style="list-style-type: none"> • atv_proj_conc(fi): Quantidade de demandas de projetos concluídas que passaram pela atividade fi. • atv_proj_total: Quantidade total de demandas de projetos concluídas.
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3 Taxa de aderência da manutenção ao processo

Descrição	Mede o percentual de aderência das demandas de manutenção ao processo.
Periodicidade	Trimestral
Fórmula	$adrenc_atv_manut = \sum_{i=1}^n \frac{atv_manut_conc(fi)}{atv_manut_total}$ <p>Onde:</p> <ul style="list-style-type: none"> • n: Quantidade de atividades do processo a serem analisadas. • fi: Representa uma das seguintes atividades do processo: Codificar Demanda; Revisar Tarefa; Realizar Teste; Homologar Sistema; e Implantar em Produção. • adrenc_atv_manut: Aderência das demandas de manutenção ao processo. • atv_manut_conc(fi): Quantidade de demandas de manutenção concluídas que passaram pela atividade fi. • atv_manut_total: Quantidade total de demandas de manutenção concluídas.

4 Taxa de aderência do projeto ao processo

Descrição	Mede o percentual de aderência das demandas de projeto ao processo.
------------------	---------------------------------------------------------------------

Periodicidade	Trimestral
Fórmula	$adrenc_atv_proj = \sum_{i=1}^n \frac{atv_proj_conc (fi)}{atv_proj_total}$ <p>Onde:</p> <ul style="list-style-type: none"> • n: Quantidade de atividades do processo a serem analisadas. • fi: Representa uma das seguintes atividades do processo: Codificar Demanda; Revisar Tarefa; Realizar Teste; Homologar Sistema; e Implantar em Produção. • adrenc_atv_proj: Aderência das demandas de projetos ao processo. • atv_proj_conc(fi): Quantidade de demandas de projetos concluídas que passaram pela atividade fi. • atv_proj_total: Quantidade total de demandas de projetos concluídas.

9. Sugestões de melhorias futuras

- No que diz respeito aos treinamentos das mudanças de softwares, adequar o processo ou ao protocolo TRT 000-2421/2014, ou ao Regulamento Geral (RG) do TRT da 13ª Região. Atualmente essa responsabilidade seria concomitante da Escola Judicial (art. 66 e parágrafos seguintes do RG), da SEGEPE e da SETIC, mais especificamente do Setor de Treinamento de Usuários da CAU (art. 185 do RG). No futuro, esta responsabilidade deveria ser atribuída apenas à SEGEPE ou à Escola Judicial, sendo retirada da SETIC.
- Melhorar o processo de criação e revisão dos manuais de usuário final.
- A integração a atividade treinamento dos softwares e implantação destes em produção com a Assessoria de Comunicação Social.
- Transferir integralmente para a CSUAP, a CAU ou a um setor a ser criado posteriormente a gestão de incidentes, tanto de softwares jurídicos quanto

administrativos, permitindo, inclusive, ao responsável por essa gestão a execução direta de comandos DML (Linguagem de Manipulação de Dados, do inglês, *Data Manipulation Language*) no banco de dados do ambiente de produção.

- Incluir, no processo de desenvolvimento, garantias para a aplicação das diretrizes de segurança de software.
- Em vez de priorizar todas as demandas no PDTIC, criar um critério objetivo que facilite a priorização das demandas pela CDMS, deixando no PDTIC apenas as demandas mais críticas e essenciais, ou seja, o mínimo. Esse critério definirá, por exemplo, quem têm mais prioridade, se as demandas judiciais ou administrativas, se as do Tribunal ou das Varas, etc. Esta melhoria futura precisará ser alinhada ou integrada com os processos de Gestão de Mudança e de Portifólio.
- Integrar ou ajustar o Processo de Desenvolvimento de Software ao Processo de Portifólio.
- Melhorar a integração entre as tarefas "Analisar demanda", "Alimentar o backlog" e "Priorizar e delegar tarefas" e o subprocesso "Iniciar Projeto".
- Integrar o Processo de Desenvolvimento de Software ao de aquisição de um novo software por licitação, terceirização.
- Montar o ambiente de homologação somente após a demanda passar pelo ambiente de teste. Hodiernamente ambos ambientes são montados no mesmo momento.
- Implementar o indicador "Percentual de Bugs em Funcionalidades Homologadas", que mede o número de *bugs* emandos de funcionalidades homologadas pelo Usuário (Cliente).

10. Referências

[1] ITIL V3 Library – Edição 2011

[2] Glossário ITIL® de Português do Brasil, v1.0, 29 de julho de 2011

[3] R., Leandro M. T. *Tipos de manutenção de Software*. Disponível em: <<http://www.leandromtr.com/tecnologia-informacao/tipos-de-manutencao-de-software/>>.

Acessado em: 05.abr.2016.

[4] SPINOLA, Rodrigo. *Manutenção de Software: Definições e Dificuldades - Artigo Revista SQL Magazine 86*. Disponível em: <<http://www.devmedia.com.br/manutencao-de-software-definicoes-e-dificuldades-artigo-revista-sql-magazine-86/20402>>. Acessado em: 05.abr.2016.

[5] BRASIL. *Regimento Geral do Tribunal Regional do Trabalho da 13ª Região*. Disponível em: <http://www.trt13.jus.br/institucional/legislacao-interna/regulamento-geral/regulamento-geral-atualizado/at_download/file>. Acessado em: 06.abr.2016.

[6] Norma ISO 12207.

[7] Norma ISO 14764.

11. Histórico de versões

Data de atualização	Versão	Alterações realizadas	Responsável
14/04/2016	0.1	Minuta inicial do documento.	Davi Cabral
15/04/2016	0.1	Reunião de levantamento de requisitos.	Luis Fabiano e Davi Cabral
19/04/2016	0.2	Minuta alterada com base nas sugestões feitas pela CDMS em reuniões realizadas até o dia 15/04/2016.	Davi Cabral
22/04/2016	0.2	Reunião de revisão e levantamento de requisitos.	Luis Fabiano e Davi Cabral
27/04/2016	0.3	Minuta alterada com base nas sugestões feitas pela CDMS em reuniões realizadas até o dia 22/04/2016.	Davi Cabral
06/06/2016	0.3	Reunião de revisão e levantamento de requisitos.	Luis Fabiano e Davi Cabral
22/06/2016	0.4.1	Minuta alterada com base nas sugestões feitas pela CDMS na reunião realizada na semana do dia 06 de junho de 2016.	Davi Cabral
29/08/2016	0.4.1	Revisão de homologação.	Samuelson e Wagner
31/08/2016	0.4.2	Minuta alterada com base nas sugestões das reuniões de homologação realizada no dia 29 de agosto de 2016.	Davi Cabral
04/11/2016	0.5	Minuta versão final	Davi Cabral